

Wright State University

CORE Scholar

---

Computer Science & Engineering Syllabi

College of Engineering & Computer Science

---

Spring 2007

## CS 480/680: Comparative Languages

Krishnaprasad Thirunarayan

Wright State University - Main Campus, [t.k.prasad@wright.edu](mailto:t.k.prasad@wright.edu)

Follow this and additional works at: [https://corescholar.libraries.wright.edu/cecs\\_syllabi](https://corescholar.libraries.wright.edu/cecs_syllabi)



Part of the [Computer Engineering Commons](#), and the [Computer Sciences Commons](#)

---

### Repository Citation

Thirunarayan, K. (2007). CS 480/680: Comparative Languages. .  
[https://corescholar.libraries.wright.edu/cecs\\_syllabi/403](https://corescholar.libraries.wright.edu/cecs_syllabi/403)

This Syllabus is brought to you for free and open access by the College of Engineering & Computer Science at CORE Scholar. It has been accepted for inclusion in Computer Science & Engineering Syllabi by an authorized administrator of CORE Scholar. For more information, please contact [library-corescholar@wright.edu](mailto:library-corescholar@wright.edu).

# CS 480/680 Comparative Languages

---

- **Instructor** : T. K. Prasad
- **Phone No.** : (937)-775-5109
- **Email** : [t.k.prasad@wright.edu](mailto:t.k.prasad@wright.edu)
- **Home page**: <http://www.cs.wright.edu/~tkprasad/>
- **Quarter** : Spring, 2007
- **Class Hrs** : TTh, 2:15 - 3:30pm, 154 RC
- **Office Hrs** : TTh, 1:40 - 2:10pm, 395 JC (or by appt.)

---

## Course Description

This course will introduce fundamental concepts and paradigms underlying the design of modern programming languages. For concreteness, we study the details of an object-oriented language (e.g. Java), and a functional language (e.g., Scheme). The overall goal is to enable comparison and evaluation of existing languages. The programming assignments will be coded in Java 5 and in Scheme.

---

## Prerequisites

- Data Structures and Algorithms. (Equivalently, CS400/600.)
  - Experience with programming in imperative languages such as C/C++, Pascal, or Ada.
- 

## Course Text and Material

1. On-line Lecture Notes.
2. K. Arnold, J. Gosling, and D. Holmes: The Java Programming Language. Addison-Wesley Publishing Co., 4th Edition, 2005. ISBN 0-321-34980-6

## References

1. Michael L. Scott, Programming Language Pragmatics. Morgan Kaufmann Publishers, 2nd Edition, 2006. ISBN 0126339511
2. [The Java Tutorial](#)
3. Ravi Sethi, Programming Languages: Concepts and Constructs. Addison-Wesley Publishing Co., 2nd Edition, 1996. ISBN 0-201-59065-4
4. R. Kent Dybvig, [The Scheme Programming Language](#), 3rd Edition. Prentice Hall, 2003.
5. [Scheme : Language Reference Manual](#)
6. [Chez Scheme Download Site \(http://www.scheme.com\)](http://www.scheme.com)

7. [DrScheme Download Site \(http://www.drscheme.org/\)](http://www.drscheme.org/)
8. [Jython Home Page](#)
9. [Dive into Python](#)

## Relevant Websites

- [Sun's Java Page](#)
  - [Java 5.0 Core APIs](#)
- [The Teaching About Programming Languages Project](#)

## Download Sites

1. JDK 1.5 (<http://java.sun.com/j2se/1.5.0/download.jsp>)
2. JBuilder 2005 Enterprise 30-Trial and Foundation is the same download ([http://www.borland.com/products/downloads/download\\_jbuilder.html](http://www.borland.com/products/downloads/download_jbuilder.html))
3. NetBeans 4.0 (<http://www.netbeans.org/community/releases/40/index.html>)
4. Eclipse 3.0 (<http://www.eclipse.org/downloads/index.php>)
5. TextPad Editor ([www.textpad.com](http://www.textpad.com))
6. WinZip ([www.winzip.com](http://www.winzip.com))
7. Apache Tomcat ([www.apache.org](http://www.apache.org))

## Java IDE Tutorials by Y. Daniel Liang

1. [Compiling and Running Java from the Command Window](#)
2. [Compiling and Running Java from TexPad](#)
3. [JBuilder Tutorial](#)
4. [NetBeans Tutorial](#)
5. [Eclipse Tutorial](#)

## Course Load

The course load includes a mix of homeworks and programming assignments worth 30 points, a midterm worth 30 points and a final worth 40 points. Normally, CS680 students are assigned additional homework problems and are expected to solve additional/different problems in the tests.

## Grading

The letter grades will be assigned using the following scale: A[90-100], B[80-90), C[70-80), D[60-70), and F[0-60). However, I reserve the right to adjust the scale somewhat to utilize the gaps in the distribution. Academic dishonesty will be "rewarded" with a grade of "F". "Sharing/reuse" of solutions to assignment problems is strictly prohibited.

## Attendance Policy

All registered students are expected to attend all lectures. In case a student is absent from a lecture due to unavoidable circumstances, the student is still responsible for the material covered in the class, as it is typically available from the course web-page well in advance. Furthermore, the student is expected to find out about in-class announcements from their colleagues/instructor.

## Class Schedule and Syllabus

### Topic

- Class 1** Evolution of Programming Languages
- Class 2** Syntax Specification : Grammars
- Class 3** Object-Oriented Programming
- Class 4** Java Design Goals
- Class 5** Types, Values, Variables
- Class 6** Arrays; Classes
- Class 7** Inheritance; Polymorphism
- Class 8** Interfaces; Packages; Strings
- Class 9** **Midterm (April 24)**
- Class 10** Exceptions
- Class 11** Threads
- Class 12** (continue) (Scripting vs Systems PL)
- Class 13** Symbolic Data; List Processing
- Class 14** Styles : Functional vs Procedural
- Class 15** Recursive Definitions ( Scheme-Startup )(Examples)
- Class 16** Abstraction : Higher Order Functions
- Class 17** Scoping; Closures
- Class 18** Scheme Interpreter
- Class 19** (continue)
- Class 20** (continue)
- Class \*** Parameter Passing Mechanisms
- Class \*** Implementing Subprograms
- Finals (June ??)**

## Assignments (Spring 2007)

- [Assignment 1](#)
- [Assignment 2](#)

---

## Exams (Spring 2006)

- [Midterm](#)
- [Final](#)

---

[T. K. Prasad](#) (04/02/07 05:43:30 PM )